AF $ IFW

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Application No.	09/541,444
Filing Date	March 31, 2000
First Named Inventor	Erik C. Cota-Robles
Art Unit	2126
Examiner Name	George L. Opie
Attorney Docket Number	42390P7920

Total Number of Pages in This Submission: 24

ENCLOSURES *(check all that apply)*

☒ Fee Transmittal Form
- ☒ Fee Attached

☐ Amendment / Response
- ☐ After Final
- ☐ Affidavits/declaration(s)

☐ Extension of Time Request

☐ Express Abandonment Request

☐ Information Disclosure Statement
- ☐ PTO/SB/08

☐ Certified Copy of Priority Document(s)

☐ Response to Missing Parts/ Incomplete Application
- ☐ Basic Filing Fee
- ☐ Declaration/POA
- ☐ Response to Missing Parts under 37 CFR 1.52 or 1.53

☐ Drawing(s)

☐ Licensing-related Papers

☐ Petition

☐ Petition to Convert a Provisional Application

☐ Power of Attorney, Revocation Change of Correspondence Address

☐ Terminal Disclaimer

☐ Request for Refund

☐ CD, Number of CD(s)

☐ After Allowance Communication to Group

☐ Appeal Communication to Board of Appeals and Interferences

☒ Appeal Communication to Group (Appeal Notice, Brief, Reply Brief)

☐ Proprietary Information

☐ Status Letter

☒ Other Enclosure(s) *(please identify below)*:

-Check in the amount of $340.00
-Return Receipt Postcard

Remarks

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm *or* Individual name	Gregory D. Caldwell, Reg. No. 39,926 BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP
Signature	
Date	

CERTIFICATE OF MAILING/TRANSMISSION

I hereby certify that this correspondence is being deposited with the United States Postal Service on the date shown below with sufficient postage as first class mail in an envelope addressed to: Mail Stop Appeal Brief-Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Typed or printed name	Rachael Brown		
Signature		Date	December 7, 2004

Based on PTO/SB/21 (04-04) as modified by Blakely, Solokoff, Taylor & Zafman (wlr) 06/04/2004.
SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450

FEE TRANSMITTAL for FY 2004

Effective 01/01/2004. Patent fees are subject to annual revision.

☐ Applicant claims small entity status. See 37 CFR 1.27.

TOTAL AMOUNT OF PAYMENT	($)	340.00

Complete if Known	
Application Number	09/541,444
Filing Date	March 31, 2000
First Named Inventor	Erik C. Cota-Robles
Examiner Name	George L. Opie
Art Unit	2126
Attorney Docket No.	42390P7920

METHOD OF PAYMENT *(check all that apply)*

☒ Check ☐ Credit card ☐ Money Order ☐ Other ☐ None
☐ Deposit Account

Deposit Account Number	02-2666
Deposit Account Name	Blakely, Sokoloff, Taylor & Zafman LLP

The Commissioner is authorized to: *(check all that apply)*

☐ Charge fee(s) indicated below ☒ Credit any overpayments

☒ Charge any additional fee(s) or underpayment of fees as required under 37 CFR §§ 1.16, 1.17, 1.18 and 1.20.

☐ Charge fee(s) indicated below, **except for the filing fee** to the above-identified deposit account

FEE CALCULATION

1. BASIC FILING FEE

Large Entity		Small Entity			
Fee Code	Fee ($)	Fee Code	Fee ($)	Fee Description	Fee Paid
1001	790	2001	395	Utility filing fee	
1002	350	2002	175	Design filing fee	
1003	550	2003	275	Plant filing fee	
1004	790	2004	395	Reissue filing fee	
1005	160	2005	80	Provisional filing fee	

SUBTOTAL (1) ($)

2. EXTRA CLAIM FEES

			Extra Claims		Fee from below		Fee Paid
Total Claims		- 20** =		X		=	
Independent Claims		- 3 =		X		=	
Multiple Dependent						=	

Large Entity		Small Entity		
Fee Code	Fee ($)	Fee Code	Fee ($)	Fee Description
1202	18	2202	9	Claims in excess of 20
1201	88	2201	44	Independent claims in excess of 3
1203	300	2203	150	Multiple Dependent claim, if not paid
1204	88	2204	44	**Reissue independent claims over original patent
1205	18	2205	9	**Reissue claims in excess of 20 and over original patent

SUBTOTAL (2) ($)

***or number previously paid, if greater, For Reissues, see below*

FEE CALCULATION (continued)

3. ADDITIONAL FEES

Large Entity		Small Entity			
Fee Code	Fee ($)	Fee Code	Fee ($)	Fee Description	Fee Paid
1051	130	2051	65	Surcharge - late filing fee or oath	
1052	50	2052	25	Surcharge - late provisional filing fee or cover sheet.	
2053	130	2053	130	Non-English specification	
1812	2,520	1812	2,520	For filing a request for *ex parte* reexamination	
1804	920*	1804	920*	Requesting publication of SIR prior to Examiner action	
1805	1,840*	1805	1,840*	Requesting publication of SIR after Examiner action	
1251	110	2251	55	Extension for reply within first month	
1252	430	2252	215	Extension for reply within second month	
1253	980	2253	490	Extension for reply within third month	
1254	1,530	2254	765	Extension for reply within fourth month	
1255	2,080	2255	1,040	Extension for reply within fifth month	
1404	340	2401	170	Notice of Appeal	
1402	340	2402	170	Filing a brief in support of an appeal	340.00
1403	300	2403	150	Request for oral hearing	
1451	1,510	2451	1,510	Petition to institute a public use proceeding	
1452	110	2452	55	Petition to revive - unavoidable	
1453	1,370	2453	685	Petition to revive - unintentional	
1501	1,370	2501	685	Utility issue fee (or reissue)	
1502	490	2502	245	Design issue fee	
1503	660	2503	330	Plant issue fee	
1460	130	2460	130	Petitions to the Commissioner	
1807	50	1807	50	Processing fee under 37 CFR 1.17(q)	
1806	180	1806	180	Submission of Information Disclosure Stmt	
8021	40	8021	40	Recording each patent assignment per property (times number of properties)	
1809	790	1809	395	Filing a submission after final rejection (37 CFR § 1.129(a))	
1810	790	2810	395	For each additional invention to be examined (37 CFR § 1.129(b))	
1801	790	2801	395	Request for Continued Examination (RCE)	
1802	900	1802	900	Request for expedited examination of a design application	
Other fee (specify)					

* Reduced by Basic Filing Fee Paid

SUBTOTAL (3) ($) 340.00

SUBMITTED BY				Complete (if applicable)	
Name *(Print/Type)*	Gregory D. Caldwell	Registration No. (Attorney/Agent)	39,926	Telephone	(503) 439-8778
Signature				Date	12/07/04

Based on PTO/SB/17 (10-03) as modified by Blakely, Solokoff, Taylor & Zafman (wlr) 02/10/2004.
SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re patent application of:

Erik C. Cota-Robles, et al.

Serial No.:	09/541,444	Group Art Unit:	2126
Filed:	March 31, 2000	Examiner:	George L. Opie
FOR:	**REAL-TIME SCHEDULING OF VIRTUAL MACHINES**		

APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

Applicant (hereinafter Appellant) submits this appeal brief, thus perfecting the notice of appeal filed on October 8, 2004.

The required headings and subject matter follow.

(i) Real party in interest.

This case is assigned of record to Intel Corporation, who is the real party in interest.

(ii) Related appeals and interferences.

There are no known related appeals and / or interferences.

(iii) Status of claims.

Claims 1-30 are pending in the case, claims 9 and 30 are objected to, and claims 1-8 and 10-29 stands rejected. The rejections of claims 1-8 and 10-29 are being appealed.

(iv) Status of amendments.

No amendments were filed subsequent to the final rejection.

(v) Summary of claimed subject matter.

Claims 1, 10, and 27 require scheduling a plurality of virtual machines based, at least in part, on respective resource requirement and interrupt periods. Claims 1, 10, and 27 includes determining a respective resource requirement for each virtual machine; determining a respective interrupt period (page 6, lines 12-15, Fig. 4, one of OS1 425, WIN32 API 430, user application 1 450 and application 2 455, resource management 475) for each virtual machine; and scheduling (page 6, lines 25-28, Fig. 4, block 415) a plurality of virtual machines based, at least in part, on the resource requirement and interrupt period. The computing platform being adapted to implement, at least, a virtual machine monitor (VMM 415, Fig.4) and a plurality of virtual machines (virtual machine A 420 and virtual machine B 435 of Fig. 4, or virtual machine A 140 and virtual machine B 150 of Fig. 1), the virtual machine monitor (VMM 415 of Fig. 4, or VMM comprising H/W Interrupt virtualizer 120, Interrupt period detector 130, and VMM scheduler 180 of Fig. 1) being capable of scheduling the virtual machines to execute real-time applications based, at least in part, on a resource requirement for each virtual machine and an interrupt period for each virtual machine.

Claims 2-3, 5-6, and 11-13 as described in page 5-7 of the application require determining resource requirement and/or interrupt period requirement and communicating the resource requirement and the interrupt requirement from an operating system running within the virtual machine and an application running within the operating system running within the virtual machine.

Claims 4, 7, 14, and 28 require dynamically maintaining values for a resource requirement and an interrupt period in a resource management application. In page 6, lines 11-28 of the application, some embodiments such as resource management application 475 and 480 may dynamically maintain values for a resource requirement and an interrupt period based on which plurality of virtual machines may be scheduled.

Claim 8, 15-16, and 29 require determining the respective resource requirement based on monitoring whether a virtual machine reaches an idle loop, increasing respective resource requirement if the idle loop is not reached, and decreasing the respective resource requirement if the idle loop is reached before a pre-determined percentage of the resource requirement has been utilized. In page 8, line 11 through page 9, line 9 some embodiments determine using a state machine 200, the respective resource requirement based on monitoring whether a virtual machine reaches an idle loop, increasing respective resource requirement if the idle loop is not reached, and decreasing the respective resource requirement if the idle loop is reached before a pre-determined percentage of the resource requirement has been utilized.

Claims 17-22 require initializing the interrupt period values, generating virtualized interrupts by virtualizing hardware interrupts, filtering known non-periodic interrupts, rejecting detected aperiodic interrupts, and adjusting said interrupt period values iteratively until substantially equivalent to actual interrupt periods. In page 7, line 23 through page 8, line 18 of the application, some embodiments may include an interrupt period detector to determine periods for periodic interrupts. Some embodiments may further comprise an interrupt virtualizer that filters known non-periodic interrupts such as, for example, interrupts associated with a keyboard from the interrupt period detector. Furthermore, the interrupt period detector may reject detected aperiodic interrupts not filtered by the interrupt virtualizer. The interrupt period detector may then provide a scheduler with the determined periods for the periodic interrupts so that the scheduler may schedule servicing of the periodic interrupts accordingly.

(vi) Grounds of rejection to be reviewed on appeal.

I. Claims 1-3, 5-6, 10-13, and 27 stand rejected under 35 U.S.C. § 102(e), as being anticipated by US Patent No. 6,421,702 (Gulick).

II. Claims 4, 7-8, 14-16, and 28-29 stand rejected under 35 U.S.C. § 103, as being unpatentable over US Patent No. 6,421,702 (Gulick) in view of US Patent No. 6,092,095 (Maytal).

III. Claims 17-26 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,412,035 (Webber).

(vii) Argument.

I. Claims 1-3, 5-6, 10-13, and 27 stand rejected under 35 U.S.C. § 102(e), as being anticipated by US Patent No. 6,421,702 (Gulick) is in error and should be reversed.

As is well-established, in order to successfully assert a *prima facie* case of anticipation, the Official Action must provide a single prior art document that includes every element and limitation of the claim or claims being rejected. Therefore, if even one element or limitation is missing from the cited document, the Official Action has not succeeded in making a prima facie case.

Claims 1, 10, and 27

Claims 1, 10 and 27 require scheduling a plurality of ***virtual machines***. The Examiner appears to be relying on the description provided in column 7, line 48 through column 8, line 5 of Gulick (US patent No. 6,421,702) for a teaching of the above limitations of claims 1, 10 and

27. However, Gulick merely teaches a scheduling of ***isochronous tasks***. As indicated by the Appellant in a response filed on January 28, 2004, the Gulick patent does not contain a single reference to a virtual machine. Gulick simply appears to provide no teaching with respect to virtual machines let alone to scheduling of virtual machines as required by the invention of claims 1, 10 and 27.

In response to the Appellant's position, the Examiner indicated in the Official Action mailed June 15, 2004 that Gulick discloses an operating system scheduling interrupt periods for different types of real-time and general purpose processing. The Examiner further stated that he has a duty and responsibility to the public and the Appellant to interpret the claims as broadly as **reasonably** possible during prosecution. The Examiner also stated that the Appellant uses terminology (e.g. virtual machine) that has a broad meaning in the art, and thus requires a broad interpretation.

While the Appellant does not disagree that the Appellant has used broad terminology to claim the Appellant's invention, Appellant believes the Examiner has given the term "virtual machine" an unreasonably broad interpretation in the subject claim. A "virtual machine" is a well defined term in the art. Simply put those skilled in the art would understand that a "virtual machine" has characteristics and other aspects which distinguish the term from a "task". A virtual machine is such a well understood term that even non-technical dictionaries provide a reasonable definition of a virtual machine. For example, The American Heritage® Dictionary of the English Language, Fourth Edition, Copyright © 2000 defines a virtual machine as "a software program that emulates a hardware system" which is in line with the Background of Appellant's application which describes a virtual machine as software that emulates hardware.

In contrast, Gulick provides no teaching in regard to virtual machines or software programs that emulate hardware. Gulick as stated above merely teaches scheduling of tasks and in particular scheduling of isochronous tasks. Accordingly, Examiner must be taking the position that a "task" and a "virtual machine" are the same thing. Ironically, the term "task" is not as well defined in the art and could have several different meanings such as program, thread, process, job, etc. While these terms have slightly different connotations in different environments, none of these different connotations refer to "software that emulates hardware" or to "virtual machines". In other words, those skilled in the art would not interpret a virtual machine so broadly as to equate a task with a virtual machine.

The Examiner correctly stated that each term should be given its broadest reasonable interpretation and that limitations in the specification cannot be read into the claims. However, at the same time, terms of the claims cannot be interpreted so broadly as to divorce them of their plain and ordinary meaning. Appellant simply requests that the term "virtual machine" be interpreted in accordance with its plain and ordinary meaning. Under such an interpretation, Gulick does not anticipate the invention of Appellant's claims 1, 10 and 27 since Gulick provides no teaching in regard to "virtual machines".

Claims 2-3 and 5-6

Claims 2-3 and 5-6 require determining a resource requirement and/or an interrupt period requirement and communicating the resource requirement and the interrupt requirement from an operating system running within the virtual machine. Examiner appears to be relying on Gulick column 3, lines 40-51 for such limitations. Gulick, however, appears to teach executing pending isochronous tasks in a non-real time operating system in response to a time-slice interrupt, and executing a non-isochronous task if no additional isochronous tasks are pending. Gulick appears to provide no teaching with respect to determining resource requirement and/or interrupt period and communicating the determined resource requirement and/or interrupt period values from ***the operating system running within the virtual machine*** as required by claims 2-3 and 5-6. Appellant respectfully requests the rejection of claims 2-3 and 5-6 be reversed.

Claims 11-13

Claims 11-13 require determining a resource requirement and/or an interrupt period requirement and communicating the resource requirement and the interrupt requirement from an application running within the operating system running within the virtual machine. Again, the Examiner appears to be relying on Gulick column 3, lines 40-51 for a teaching of such limitations. As stated above, Gulick appears to teach executing pending isochronous tasks in a non-real time operating system in response to a time-slice interrupt, and executing a non-isochronous task if no additional isochronous tasks are pending. Gulick appears to provide no teaching with respect to determining resource requirement and/or interrupt period and communicating the determined resource requirement and/or interrupt period values from ***an application running within a virtual machine*** as required by claims 11-13. Appellant respectfully requests the rejection of claims 11-13 be reversed.

II. Claims 4, 7-8, 14-16, and 28-29 stand rejected under 35 U.S.C. § 103, as being unpatentable over US Patent No. 6,421,702 (Gulick) in view of US Patent No. 6,092,095 (Maytal) is in error and should be reversed.

Claims 4, 7-8, 14-16, and 28-29

Each of claims 4, 7-8, 14-16, and 28-29 depends from at least one of claims 1, 3, 6, 10 and 27. As a result, each of claims 4, 7-8, 14-16, and 28-29 is allowable for at least one or more of the reasons stated above in regard to claims 1, 3, 6, 10 and 27. Appellants respectfully request the rejection of claims 4, 7-8, 14-16, and 28-29 be reversed.

III. Claims 17-26 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,412,035 (Webber) is in error and should be reversed.

Claims 17-22:

Claims 17-22 require initializing the interrupt period values, generating virtualized interrupts by virtualizing hardware interrupts, filtering known non-periodic interrupts, rejecting detected aperiodic interrupts, and adjusting said interrupt period values iteratively until substantially equivalent to actual interrupt periods. Examiner appears to rely on description provided in page 5, lines 10-24 to teach initializing the interrupt period values, generating virtualized interrupts by virtualizing hardware interrupts, filtering known non-periodic interrupts, rejecting detected aperiodic interrupts, and adjusting said interrupt period values iteratively until substantially equivalent to actual interrupt periods. As disclosed in the Appellant's application (page 7, line 23 through page 8, line 18) some embodiments may include an interrupt period detector to determine periods for periodic interrupts. Some embodiments may further comprise an interrupt virtualizer that filters known non-periodic interrupts such as, for example, interrupts associated with a keyboard from the interrupt period detector. Furthermore, the interrupt period detector may reject detected aperiodic interrupts not filtered by the interrupt virtualizer. The interrupt period detector may then provide a scheduler with the determined periods for the periodic interrupts so that the scheduler may schedule servicing of the periodic interrupts accordingly.

Webber on the other hand describes a fast interrupt handling module that provides facilities for faster servicing of certain real-time interrupts (page 5, lines 47-67 and page 7, line 3 through page 8 line 14). In particular, Webber provides various functions for changing the interrupt frequency. However, Appellant was unable to locate where Webber discloses adjusting the interrupt period values iteratively until substantially equivalent to actual interrupt periods.

Further, Appellant was unable to locate where Webber discloses filtering of known non-periodic interrupts. In regard to filtering, it appears that Webber merely provides a Hook_Device_Services() function that has several member functions that filter corresponding functions of the VTD.386 module. These functions of the Hook_Device_Services() appear to override or filter software calls to similar functions of the VTD.386 module but as far as the Appellant can discern do not filter non-periodic interrupts. Applicant respectfully requests the rejection of claims 17-26 be reversed.

CONCLUSION

In view of the foregoing, favorable reconsideration and reversal of the rejections is respectfully requested. Early notification of the same is earnestly solicited. If there are any questions regarding the present application, the Examiner and / or the Board is invited to contact the undersigned attorney at the telephone number listed below.

Respectfully submitted,

December 7, 2004
Date

Gregory D. Caldwell
Reg. No. 39,926
(503)439-8778

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail with sufficient postage in an envelope addressed to: MS Appeal Brief, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313 on December 7, 2004.

Rachael Brown **Date: December 7, 2004**

(viii) Claims appendix.

1. A method for scheduling a plurality of virtual machines comprising:

determining a respective resource requirement for each virtual machine;

determining a respective interrupt period for each virtual machine; and

scheduling said plurality of virtual machines based, at least in part, on said respective resource requirement and interrupt period values.

2. The method of claim 1 wherein, determining said respective resource requirement and interrupt period comprises communicating said respective resource requirement and interrupt period from an operating system running within said respective virtual machine.

3. The method of claim 1 wherein, determining said resource requirement and said interrupt period comprises communicating said resource requirement and said interrupt period from an application running within an operating system running within said respective virtual machine.

4. The method of claim 3 further comprising:

dynamically maintaining values for said resource requirement and said interrupt period, wherein said application is a resource management application.

5. The method of claim 1 wherein, determining said resource requirement comprises communicating said resource requirement from an operating system running within said respective virtual machine.

6. The method of claim 5 wherein, determining said resource requirement comprises communicating said resource requirement from an application running within an operating system running within said respective virtual machine.

7. The method of claim 6, wherein said application is a resource management application, which dynamically maintains said respective resource requirement.

8. The method of claim 1, wherein determining a respective resource requirement comprises:

monitoring whether a virtual machine reaches an idle loop;

increasing said respective resource requirement if said idle loop is not reached;

decreasing said respective resource requirement if said idle loop is reached before a predetermined percentage of said resource requirement has been utilized.

9. The method of claim 8, wherein determining respective interrupt period values comprises:

filtering non-periodic interrupts;

rejecting aperiodic interrupts;

estimating said respective interrupt period values for periodic interrupts; and

converging said respective interrupt period values to be substantially equivalent to actual periods for said periodic interrupts.

10. An article comprising: a storage medium having stored thereon instructions that, when executed, result in a computing platform having the capability to:

schedule a plurality of virtual machines implemented in said computing platform based, at least in part, on a respective resource requirement and an a respective interrupt period for each virtual machine of said plurality.

11. The article of claim 10, wherein said instructions, when executed result in the capability to communicate said respective resource requirement from an application running within a virtual machine of said plurality.

12. The article of claim 11, wherein said instructions, when executed result in the capability to communicate said respective interrupt period from an application running within a virtual machine of said plurality.

13. The article of claim 10, wherein said instructions, when executed result in the capability to communicate said respective resource requirement and said respective interrupt period from an operating system running within a virtual machine of said plurality.

14. The article of claim 10, wherein said instructions, when executed result in the capability to communicate said respective resource requirement and said respective interrupt period from a resource management application running within a virtual machine of said plurality.

15. The article of claim 10, wherein said instructions, when executed result in the capability to determine said respective interrupt period by comparing an expected interrupt period with an actual interrupt period and adjusting said respective interrupt period based, at least in part, on said comparison.

16. The article of claim 10, wherein said instructions, when executed result in the capability to determine said respective resource requirement by detecting the occurrence of an idle loop within a virtual machine of said plurality and adjusting resource requirement based, at least in part, on whether said idle loop occurs.

17. A method for determining interrupt period values comprising:

initializing said interrupt period values;

generating virtualized interrupts by virtualizing hardware interrupts;

filtering known non-periodic interrupts;

rejecting detected aperiodic interrupts; and

adjusting said interrupt period values iteratively until substantially equivalent to actual interrupt periods.

18. The method of claim 17, further comprising:

acquiring resource requirement values; and

scheduling a plurality of virtual machines to achieve real-time deadlines based, at least in part, on said interrupt period values and resource requirement values.

19. The method of claim 18, wherein said resource requirement values are acquired from said plurality of virtual machines.

20. The method of claim 17, further comprising determining resource requirement values, wherein determining said resource requirement values comprises:

initializing said resource requirement values; and

adjusting said resource requirement values iteratively based, at least in part, on a determination of an occurrence of a respective predetermined instruction.

21. The method of claim 20, wherein adjusting said resource requirement values comprises:

increasing said resource requirement values if said respective predetermined instruction does not occur;

decreasing said resource requirement values if said respective predetermined instruction occurs prior to a target time; and scheduling a plurality of virtual machines based, at least in part, on said interrupt period values and said resource requirement values.

22. An article comprising: a storage medium having stored thereon instructions that, when executed, result in a computing system having the capability to:

initialize interrupt period values;

generate virtualized interrupts by virtualizing hardware interrupts;

filter known non-periodic interrupts;

reject detected aperiodic interrupts; and

adjust said interrupt period values iteratively until substantially equivalent to actual interrupt periods.

23. The article of claim 22, wherein said instructions, when executed, further result in the capability to:

acquire resource requirement values; and

schedule a plurality of virtual machines to achieve real-time deadlines based, at least in part, on said interrupt period values and resource requirement values.

24. The article of claim 23, wherein said resource requirement values are acquired from said plurality of virtual machines.

25. The article of claim 22, wherein said instructions, when executed, result in said computing platform having the further capability to:

determine resource requirement values, wherein determining said resource requirement values comprises:

initializing said resource requirement values; and

adjusting said resource requirement values iteratively based, at least in part, on a determination of an occurrence of a respective predetermined instruction.

26. The article of claim 25, wherein adjusting said resource requirement values comprises:

increasing said resource requirement values if said respective predetermined instruction does not occur;

decreasing said resource requirement values if said respective predetermined instruction occurs prior to a target time; and

scheduling a plurality of virtual machines based, at least in part, on said interrupt period values and said resource requirement values.

27. A system comprising:

a computing platform;

said computing platform being adapted to implement, at least, a virtual machine monitor and a plurality of virtual machines;

said virtual machine monitor being capable of scheduling said virtual machines to execute real-time applications based, at least in part, on a resource requirement for each virtual machine and an interrupt period for each virtual machine.

28. The system of claim 27, further comprising:

an interface capable of communicating respective resource requirement and interrupt period values for said each virtual machine to said virtual machine monitor.

29. The system of claim 27, wherein said virtual machine monitor comprises:

a feedback loop capable of determining a respective resource requirement for said each virtual machine;

a hardware interrupt virtualizer capable communicating device interrupts to said plurality of virtual machines and filtering non-periodic interrupts;

an interrupt period detector capable of determining said periods for periodic interrupts and communicating said periods to a scheduler; and

said scheduler being capable of said scheduling of said plurality of virtual machines.

30. The system of claim 29, wherein said feedback loop comprises:

a detector capable of determining whether each of said virtual machines issues a predetermined instruction and indicating said determinations to a proportional integral derivative controller;

said proportional integral derivative being capable of adjusting said respective resource requirement for said each virtual machine based, at least in part, on said determination and communicating said adjusted respective resource requirement to said scheduler.

(ix) Evidence appendix.

None.

(x) Related proceedings appendix.

None.